ONTARIO
NEURODEGENERATIVE
DISEASE RESEARCH
INITIATIVE

Data Preparation App - Reference Guide

On behalf of the Neuroinformatics and Biostatistics (NIBS) platform

This reference guide helps you install and use the ONDRI Data Preparation App, with instructions on how to run the app, its limitations, and future updates.

Contents

1 Overview

The purpose of the data preparation app is to streamline a pipeline for data exploration and cleaning, and prepare tabular data for downstream analyses such as principal component analysis (PCA), correspondence analysis (CA), and outlier analysis. The app follows ONDRI protocol for preparing data, which consists of the following steps in sequential order as outlined below:

1. Read in tabular data and associated data dictionary.
2. Perform missingness checks and removal of variables and participants.
3. Select variables for analyses and define variable types.
4. Perform transformations to recode different variable types into a disjunctive format.
5. Perform imputation to handle missing values.
6. Residualize data based on age and sex. (OPTIONAL and not added in the current version of the app)

In the NIBS curation process, the data preparation app is used after performing a standards check on the required components of a data package. Likewise, it is used before running outlier analysis on the tabular data. Overall, the data preparation app serves as a middleware between the existing ONDRI Standards App and ONDRI Outliers App.

2 Installation

1. Install R first and then RStudio. Please choose the correct installer carefully as it will depend on your computer's operating system.
2. Install Git (again please choose the correct installer carefully). During the installation process, you can leave all installation options to their default and original configuration. However, you can download Git in a different folder path if you wish.
3. Open RStudio and run the following commands in the RStudio console (the bottom left pane) to install the necessary packages if you have not already done so:

```r
install.packages("devtools")
install.packages("shiny")
install.packages("shinydashboard")
devtools::install_github("nik01010/dashboardthemes")
install.packages("shinyFiles")
install.packages("shinyWidgets")
install.packages("shinyjs")
install.packages("shinyjqui")
install.packages("shinycssloaders")
install.packages("DT")
install.packages("dplyr")
install.packages("tidyr")
install.packages("tibble")
install.packages("ggplot2")
install.packages("plotly")
install.packages("missMDA")
```

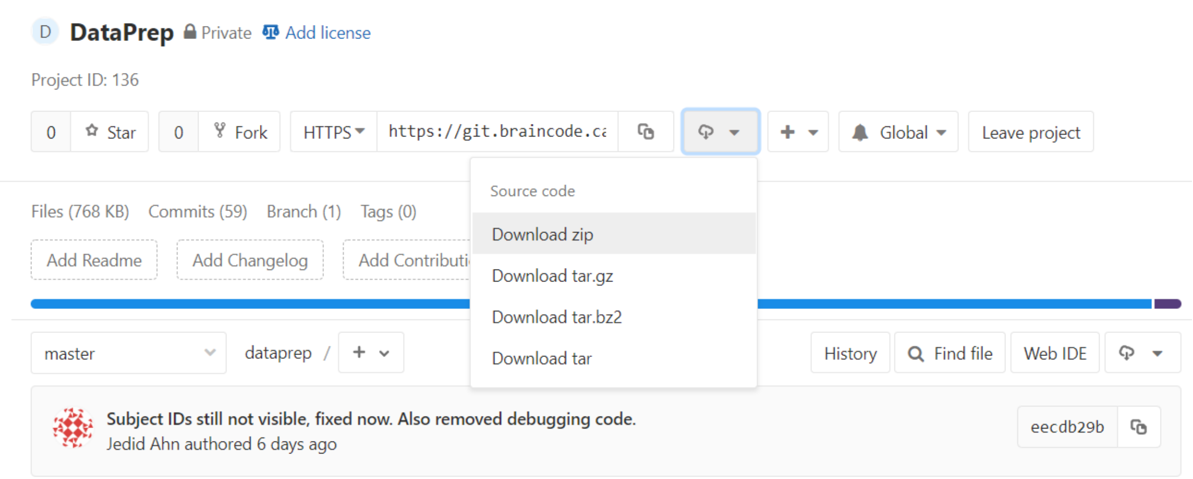
3 Deployment

There are 2 ways to deploy the app: the first way is through downloading the repository directly off of GitLab and the second way is through Git Bash. We describe both options below. Please email Derek first to request access to the repository, as it is not available to the public due to internal beta testing.

3.1 Download repository

1. Sign in to GitLab with your user account and go to the following link:
   https://git.braincode.ca/dbeaton/dataprep

2. Click the download button as shown in Figure 1, and then click "Download zip".



Subject IDs still not visible, fixed now. Also removed debugging code.
Jedid Ahn authored 6 days ago

Figure 1: Please note that the repository is private and not available on GitHub like the other two apps.

3. Browse to the directory that you saved the ZIP folder in, right click on the folder, and click "Extract All" as shown in Figure 2. Extract into a separate folder (not in the same directory as the ZIP folder). You may delete the ZIP folder afterwards.

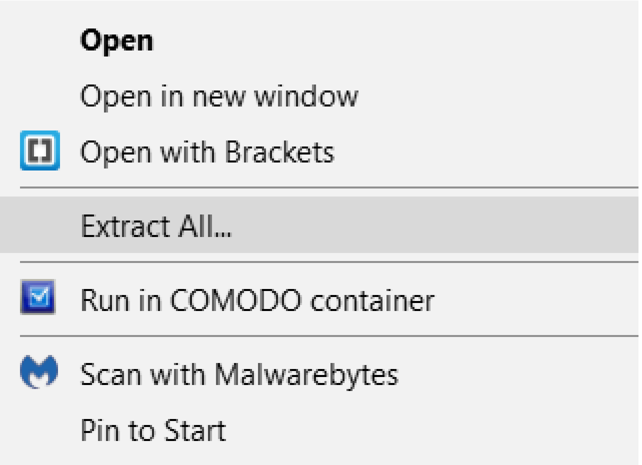


Figure 2: Please note that your drop down may look different depending on the operating system.

3.2 Git Bash

1. Right click on the folder that you wish to store the repository/program files in, and click on “Git Bash Here” as shown in Figure 3. The Git Bash application will pop up.

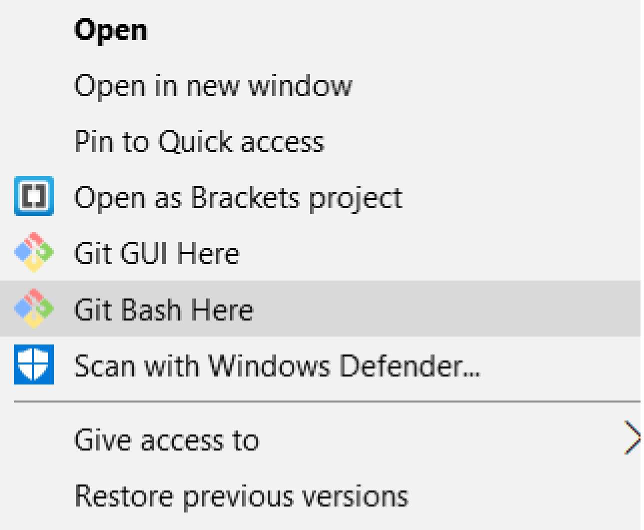
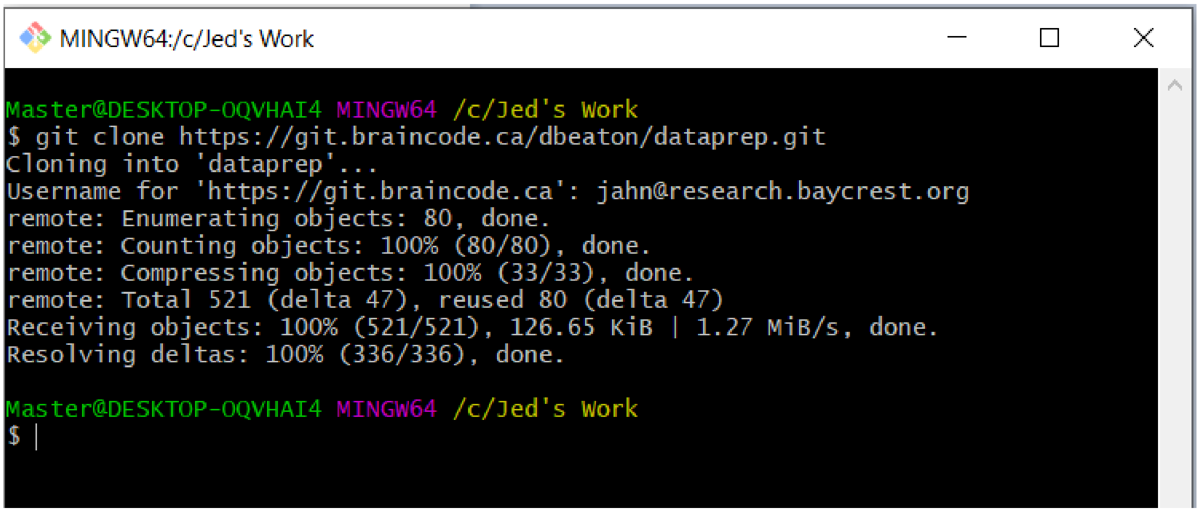


Figure 3: Please note that your drop down may look different depending on the operating system.

2. Clone the repository through Git by running the following command as shown in Figure 4:
   git clone https://git.braincode.ca/dbeaton/dataprep.git

Unlike GitHub which is public, the terminal will ask you to input your GitLab username (which is your research email address) followed by a popup window for your password.



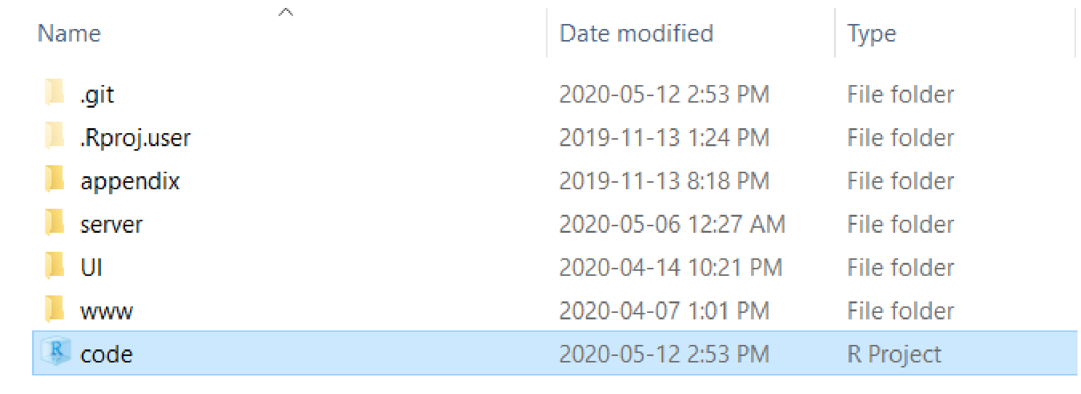
```
Master@DESKTOP-OQVHAI4 MINGW64 /c/Jed's Work
$ git clone https://git.braincode.ca/dbeaton/dataprep.git
Cloning into 'dataprep'...
Username for 'https://git.braincode.ca': jahn@research.baycrest.org
remote: Enumerating objects: 80, done.
remote: Counting objects: 100% (80/80), done.
remote: Compressing objects: 100% (33/33), done.
remote: Total 521 (delta 47), reused 80 (delta 47)
Receiving objects: 100% (521/521), 126.65 KiB | 1.27 MiB/s, done.
Resolving deltas: 100% (336/336), done.

Master@DESKTOP-OQVHAI4 MINGW64 /c/Jed's Work
$ |
```

Figure 4: The Git Bash terminal, which you can use only if you have installed Git.

3.3 Next steps after choosing 3.1 or 3.2

1. Open RStudio, go to File → Open Project, go to the "dataprep" folder within the repository, and double click code (code.Rproj) as shown in Figure 5.

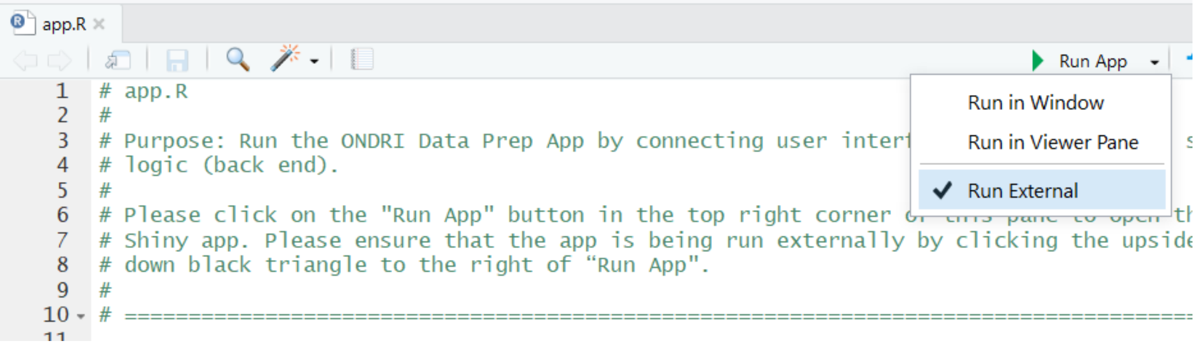


Name	Date modified	Type
.git	2020-05-12 2:53 PM	File folder
.Rproj.user	2019-11-13 1:24 PM	File folder
appendix	2019-11-13 8:18 PM	File folder
server	2020-05-06 12:27 AM	File folder
UI	2020-04-14 10:21 PM	File folder
www	2020-04-07 1:01 PM	File folder
code	2020-05-12 2:53 PM	R Project

Figure 5: code.Rproj will be evident by the RStudio symbol on the left.

2. Open app.R in the RStudio file interface.

3. Click the dropdown of "Run App" in the top right of the file interface and select "Run External" as shown in Figure 6. This will run the app through your internet browser rather than through a window. NOTE: If the app is not run externally, step #6 in the pipeline will not function properly.



```
# app.R
#
# Purpose: Run the ONDRI Data Prep App by connecting user interf
# logic (back end).
#
# Please click on the "Run App" button in the top right corner of this pane to open th
# Shiny app. Please ensure that the app is being run externally by clicking the upside
# down black triangle to the right of "Run App".
#
# ==========================================================================================
```

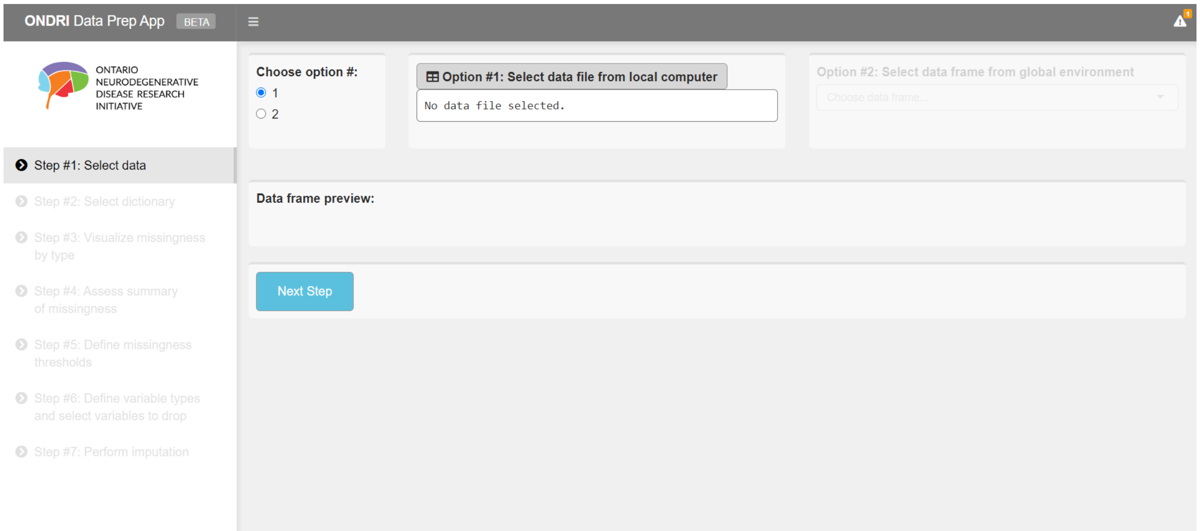
Figure 6: To open the dropdown, click the upside down black triangle to the right of "Run App".

4. Click "Run App" or type shiny::runApp() in the RStudio console.

4 Running App

IMPORTANT: Please note that the data package has to pass standards through the ONDRI Standards App before proceeding to data preparation.

The app breaks the pipeline into seven distinct and sequential steps, which are shown in a menu in the left pane of the user interface (UI). Each step is addressed below:


ONDRI Data Prep App
BETA
ONTARIO NEURODEGENERATIVE DISEASE RESEARCH INITIATIVE
Step #1: Select data
Step #2: Select dictionary
Step #3: Visualize missingness by type
Step #4: Assess summary of missingness
Step #5: Define missingness thresholds
Step #6: Define variable types and select variables to drop
Step #7: Perform imputation
Option #1: Select data file from local computer
Option #2: Select data frame from global environment
Choose data frame...
Data frame preview:
Next Step

Choose option #:
● 1
○ 2

No data file selected.

Figure 7: The initial UI when the app is opened. The seven steps of the pipeline are shown in the menu.

4.1 Select data

There are two ways to read in tabular data through the app. Option #1 allows you to select data as a csv file from your local computer. On the other hand, option #2 allows you to select a data frame from your RStudio global environment.

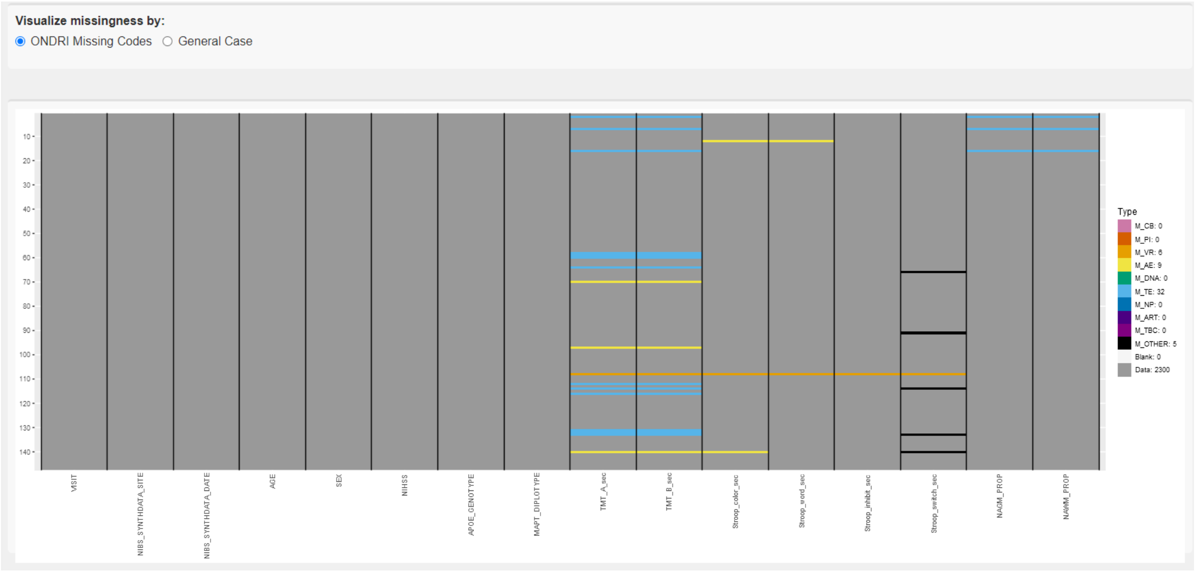
4.2 Select dictionary

Likewise, there are two ways to read in the associated data dictionary through the app. Option #1 remains file selection and option #2 remains data frame selection. However, if no dictionary exists, you can select the "Dictionary not available" option.

4.3 Visualize missingness

There are two distinct heatmaps for visualizing missingness. The first heatmap assesses missingness by ONDRI missing codes, where each missing code is represented by a distinct colour as shown in Figure 8. On the other hand, the second heatmap assesses general missingness by representing ONDRI missing codes and NA values as one common type and colour as shown in Figure 9. It is important to note that ONDRI data cannot contain any NA values, so the second heatmap represents all ONDRI missing codes through one colour only (black). In addition, both heatmaps indicate blank values through a separate colour (white), which represents conditional missingness within the data (i.e. If answer is no, please leave blank).

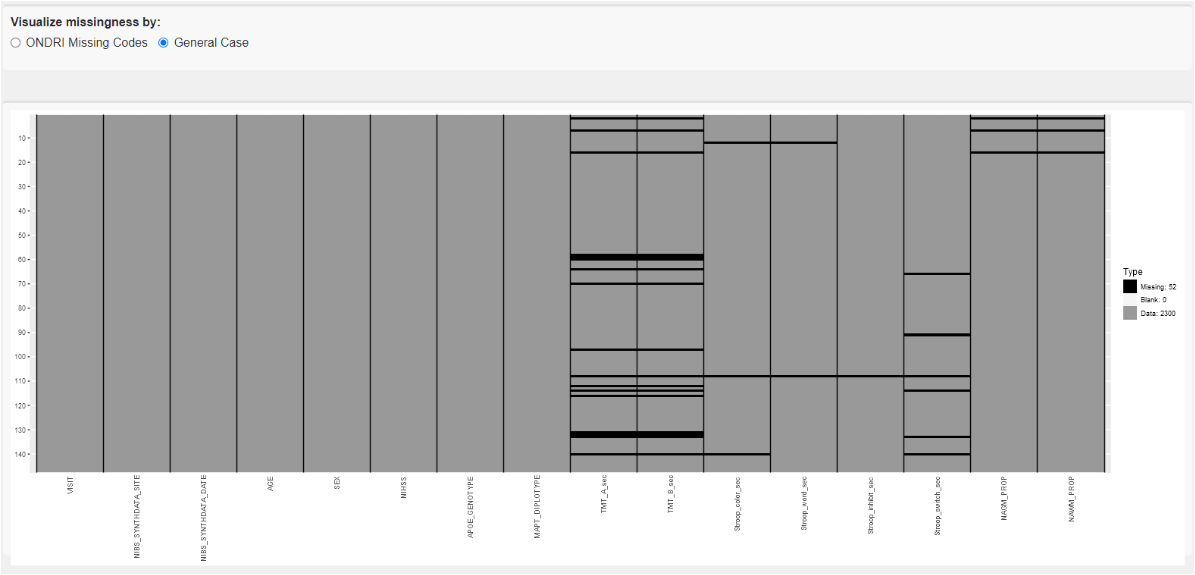
Overall, the first heatmap should only be used for ONDRI curators, while the second heatmap should only be used for general app users who are reading in non-ONDRI data.



Visualize missingness by:

● ONDRI Missing Codes ○ General Case

Figure 8: An example of an ONDRI specific missingness heatmap.



Visualize missingness by:

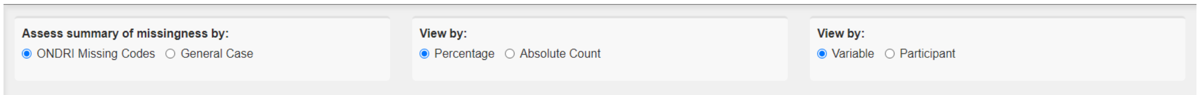
○ ONDRI Missing Codes ● General Case

Figure 9: An example of a general missingness heatmap.

4.4 Assess summary of missingness

There are a total of eight distinct tables used to summarize missingness. Once again, summary of missingness can be assessed by either ONDRI missing codes (four tables) or general missingness (another four tables).

For each scenario, missingness can be viewed through either percentage or absolute count for both variables (columns) and participants (rows). The UI is controlled through radio buttons which allow for smooth and easy navigation between tables as shown in Figure 10. Overall, this step serves as an alternative method of understanding the missingness within each variable and participant.

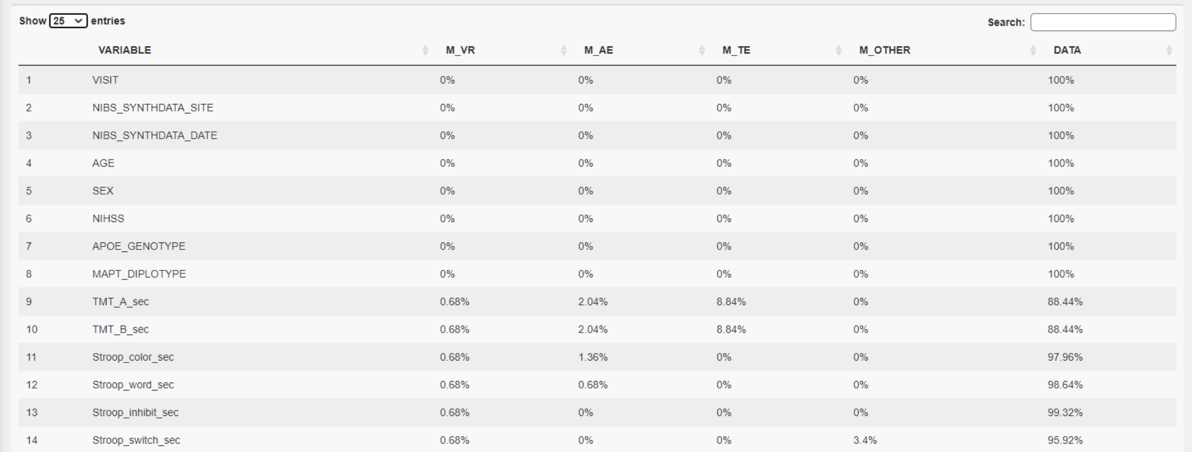


Assess summary of missingness by:
● ONDRI Missing Codes ○ General Case

View by:
● Percentage ○ Absolute Count

View by:
● Variable ○ Participant

Figure 10: The radio buttons used to control missingness table visualization.



Show 25 entries

Search:

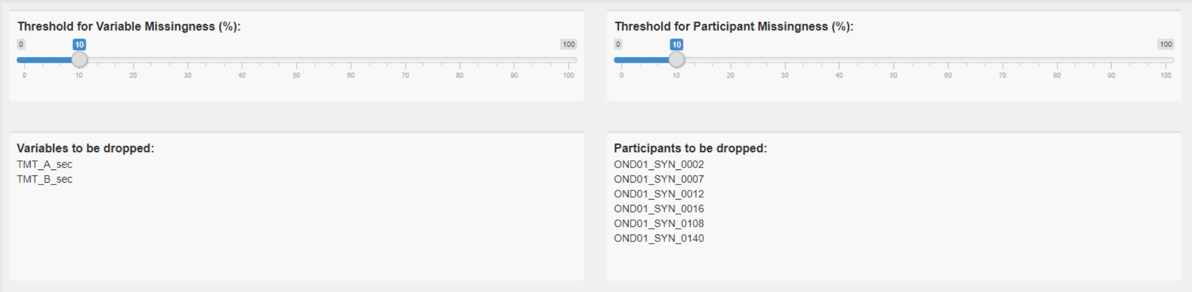
	VARIABLE	M_VR	M_AE	M_TE	M_OTHER	DATA
1	VISIT	0%	0%	0%	0%	100%
2	NIBS_SYNTHDATA_SITE	0%	0%	0%	0%	100%
3	NIBS_SYNTHDATA_DATE	0%	0%	0%	0%	100%
4	AGE	0%	0%	0%	0%	100%
5	SEX	0%	0%	0%	0%	100%
6	NIHSS	0%	0%	0%	0%	100%
7	APOE_GENOTYPE	0%	0%	0%	0%	100%
8	MAPT_DIPLOTYPE	0%	0%	0%	0%	100%
9	TMT_A_sec	0.68%	2.04%	8.84%	0%	88.44%
10	TMT_B_sec	0.68%	2.04%	8.84%	0%	88.44%
11	Stroop_color_sec	0.68%	1.36%	0%	0%	97.96%
12	Stroop_word_sec	0.68%	0.68%	0%	0%	98.64%
13	Stroop_inhibit_sec	0.68%	0%	0%	0%	99.32%
14	Stroop_switch_sec	0.68%	0%	0%	3.4%	95.92%

Figure 11: An example of a table summarizing ONDRI missingness in variables by percentage.

4.5 Define missingness thresholds

The two sliders allow for dropping of variables and participants respectively depending on their percentage of missing values. The default threshold is 10%, which drops any variables and participants containing greater than 10% missingness (in other words, less than 90% completeness). Although the sliders allow for a percentage anywhere in the range of 0% to 100%, they should ideally be somewhere between 0% and 10% for optimal downstream analyses.

Please note that the algorithm in this step is an iterative procedure that removes variables first followed by participants until the threshold in both sliders are satisfied. If the sliders get updated, the app will automatically update a list of the variables and participants that would be dropped in realtime. Once percentages are confirmed, a preview of the data frame will be provided for user viewing. To update the sliders after confirmation, please click the reset button.



Threshold for Variable Missingness (%): 10

Threshold for Participant Missingness (%): 10

Variables to be dropped:
TMT_A_sec
TMT_B_sec

Participants to be dropped:
OND01_SYN_0002
OND01_SYN_0007
OND01_SYN_0012
OND01_SYN_0016
OND01_SYN_0108
OND01_SYN_0140

Figure 12: The sliders used to control the missingness threshold and the output of variables and participants that would be dropped.

4.6 Define variable types and select variables to drop

If a dictionary was selected in Step #2, all variables will be predefined and populated by their original data types as shown in Figure 13. Otherwise, all variables will be pooled together in the topmost pane and will require drag and drop one by one as shown in Figure 14. Any variables that are irrelevant for analyses can be dropped by leaving them in the topmost pane.

Once variable types have been identified and confirmed, transformation of the data is performed automatically (and behind the scenes) so that different variable types can be recoded into a disjunctive format. In addition, a preview of the data frame after transformations will be outputted for user viewing. To update the variable data types after confirmation, please click the reset button.

NOTE: For ordinal variables, please ensure that they are coded as a numerical based ordinal scale. If the variable values are character based, please convert them to numeric and ordered values manually before running the app. Otherwise, please treat these variables as categorical.

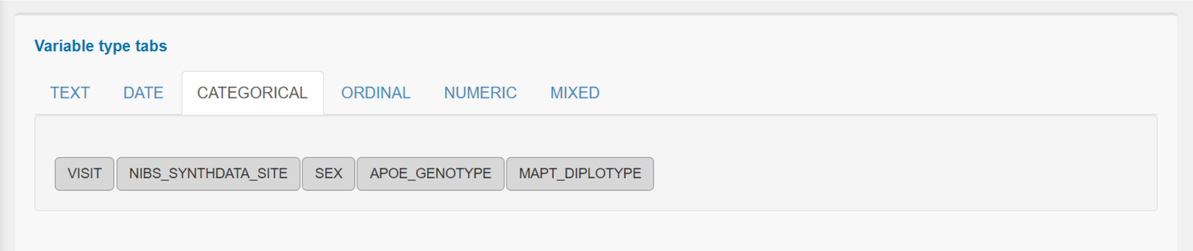


Figure 13: If a dictionary is provided, all variables will be predefined in their respective data type tabs.

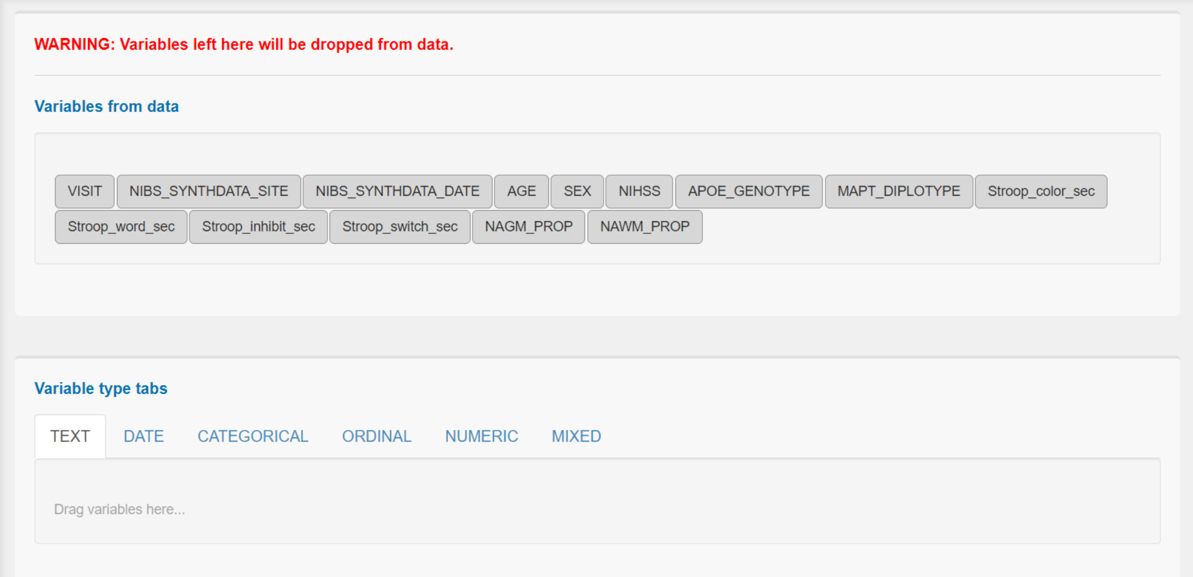

WARNING: Variables left here will be dropped from data.

Figure 14: If no dictionary is provided, all variables will be need to be defined one by one.

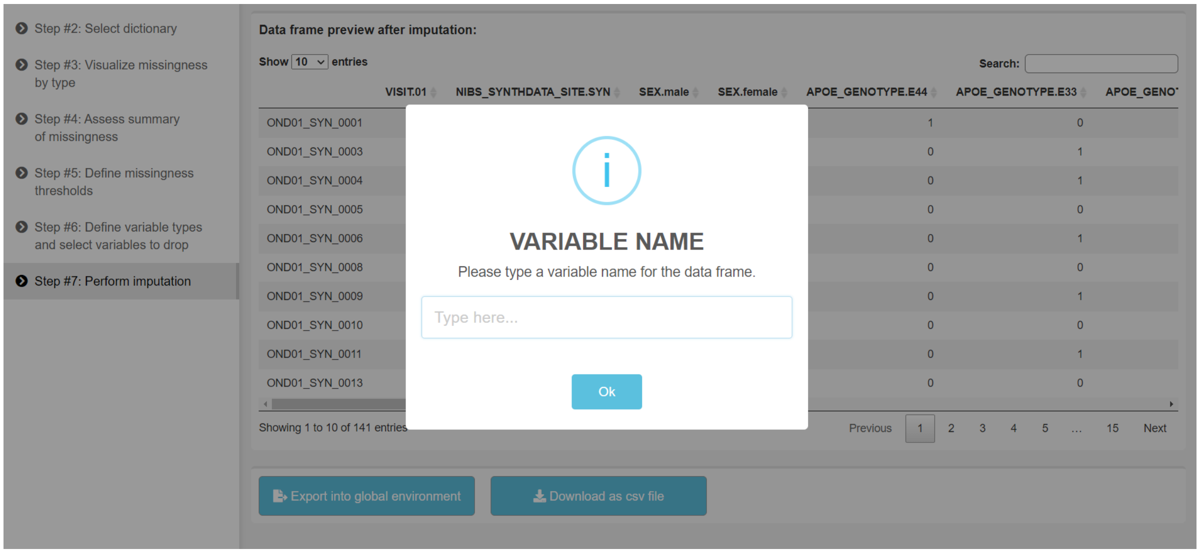
4.7 Perform imputation

Imputation represents the process of handling missing data with substituted values, and it is the last step in the pipeline. If the data as a whole is strictly continuous, select the “Strictly continuous” radio button. If the data contains a mix of continuous, categorical, and/or ordinal variables, please select the “Not strictly continuous” radio button. Once the data type is selected and confirmed, a preview of the data frame after imputation is outputted for user viewing. To update the data type after confirmation, please click the reset button.

4.8 Download new data

Once imputation is complete, the new dataset can either be exported into the RStudio global environment, or be downloaded as a csv file. Both steps are outlined below:

1. If exporting into the global environment, you will be prompted with a window asking for a variable name as shown in Figure 15. This name can only contain alphanumeric characters and underscores. **Please note that adding periods into the file name might lead to file corruption.**


Step #2: Select dictionary
Step #3: Visualize missingness by type
Step #4: Assess summary of missingness
Step #5: Define missingness thresholds
Step #6: Define variable types and select variables to drop
Step #7: Perform imputation
Show 10 entries
Search:
Previous 1 2 3 4 5 … 15 Next
Export into global environment
Download as csv file
Ok

Data frame preview after imputation:

	VISIT.01	NIBS_SYNTHDATA_SITE.SYN	SEX.male	SEX.female	APOE_GENOTYPE.E44	APOE_GENOTYPE.E33	APOE_GENO'
OND01_SYN_0001					1	0	
OND01_SYN_0003					0	1	
OND01_SYN_0004					0	1	
OND01_SYN_0005					0	0	
OND01_SYN_0006					0	1	
OND01_SYN_0008					0	0	
OND01_SYN_0009					0	1	
OND01_SYN_0010					0	0	
OND01_SYN_0011					0	1	
OND01_SYN_0013					0	0	

Showing 1 to 10 of 141 entries

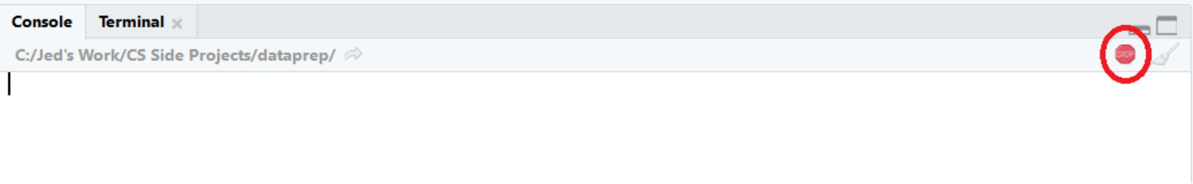
VARIABLE NAME

Please type a variable name for the data frame.

Type here...

Figure 15: The variable naming window after clicking the "Export into global environment" button.

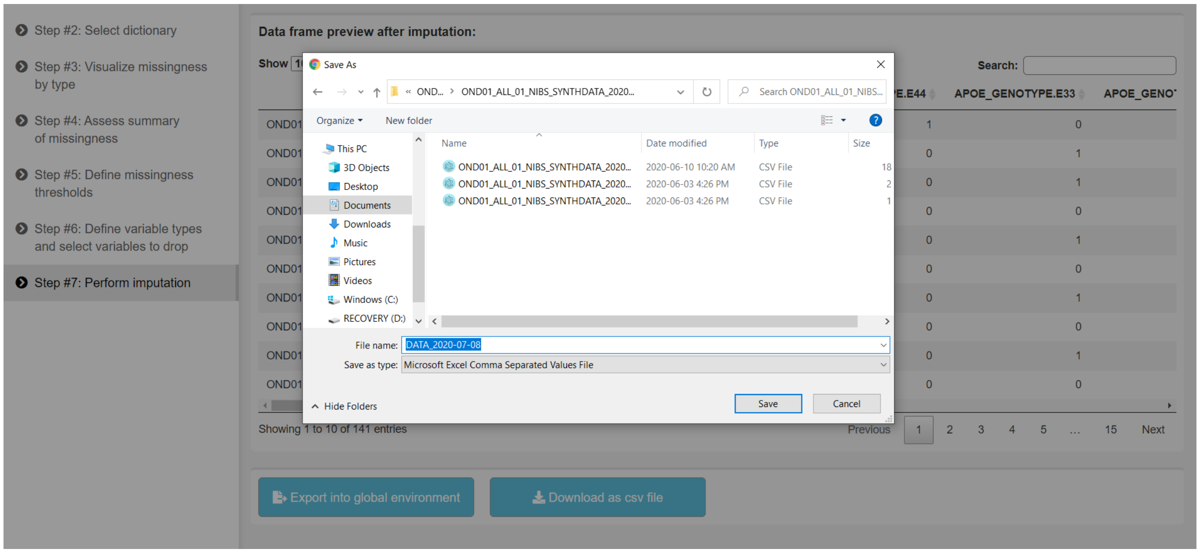
IMPORTANT: Once you receive a success message, please close the app and click the stop button in the RStudio console as shown in Figure 16. You should see the new matrix (not data frame) in your global environment in the top right pane. **Please note that this matrix contains row names, which represent the subject IDs when using ONDRI data.**



C:/Jed's Work/CS Side Projects/dataprep/

Figure 16: Please click the stop button that is circled in red once you close the app.

2. If downloading as a csv file, you will be prompted with a download window asking for download location of the file in your local computer as shown in Figure 17. **Please note that this file contains a SUBJECT variable and no row name unlike the export option.**


Step #2: Select dictionary
Step #3: Visualize missingness by type
Step #4: Assess summary of missingness
Step #5: Define missingness thresholds
Step #6: Define variable types and select variables to drop
Step #7: Perform imputation
Data frame preview after imputation:
Show 1
Search:
Save As
« OND... › OND01_ALL_01_NIBS_SYNTHDATA_2020...
Search OND01_ALL_01_NIBS...
Organize ▾
New folder
This PC
3D Objects
Desktop
Documents
Downloads
Music
Pictures
Videos
Windows (C:)
RECOVERY (D:)
Hide Folders
Save
Cancel
Previous
1
2
3
4
5
...
15
Next
Export into global environment
Download as csv file

Name	Date modified	Type	Size
OND01_ALL_01_NIBS_SYNTHDATA_2020...	2020-06-10 10:20 AM	CSV File	18
OND01_ALL_01_NIBS_SYNTHDATA_2020...	2020-06-03 4:26 PM	CSV File	2
OND01_ALL_01_NIBS_SYNTHDATA_2020...	2020-06-03 4:26 PM	CSV File	1

File name: DATA_2020-07-08

Save as type: Microsoft Excel Comma Separated Values File

	E.E44	APOE_GENOTYPE.E33	APOE_GENO
OND01	1	0	
OND01	0	1	
OND01	0	1	
OND01	0	0	
OND01	0	1	
OND01	0	0	
OND01	0	1	
OND01	0	0	
OND01	0	1	
OND01	0	0	

Showing 1 to 10 of 141 entries

Figure 17: The download window after clicking the "Download as csv file" button.

4.9 Extra notes and navigation

The left menu allows for previous steps in the pipeline to be accessed in a user-friendly manner. However, once you return to a particular step, subsequent steps may be inaccessible from the menu as changes made in the current step would affect the remainder of the pipeline. If this is the case, the pipeline from the current step onward would be reset.

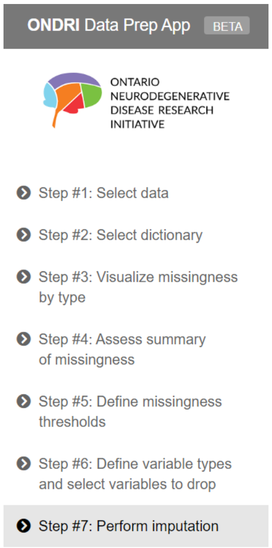

ONDRI Data Prep App
BETA
ONTARIO NEURODEGENERATIVE DISEASE RESEARCH INITIATIVE
Step #1: Select data
Step #2: Select dictionary
Step #3: Visualize missingness by type
Step #4: Assess summary of missingness
Step #5: Define missingness thresholds
Step #6: Define variable types and select variables to drop
Step #7: Perform imputation

Figure 18: This menu is used to return to previous steps in the pipeline.

5 Program Limitations

To come.

6 Development, issues, and updates

As the app is currently undergoing internal beta testing, patches will be pushed to GitLab as bugs and suggestions come in. Curators, RAs, and students will be notified when major updates become available.

If you chose to download the folder, simply download the folder again. If you installed Git, simply pull from the existing folder through Git Bash by running the following line: **git pull origin master**

For any bug reports or enhancement suggestions, please add an issue to the GitLab repository so they can be tracked. Please click the “Issues” button in the menu to create a new issue, assign Jedid Ahn, select no milestone, and label it as either a bug (red) or an enhancement (green).

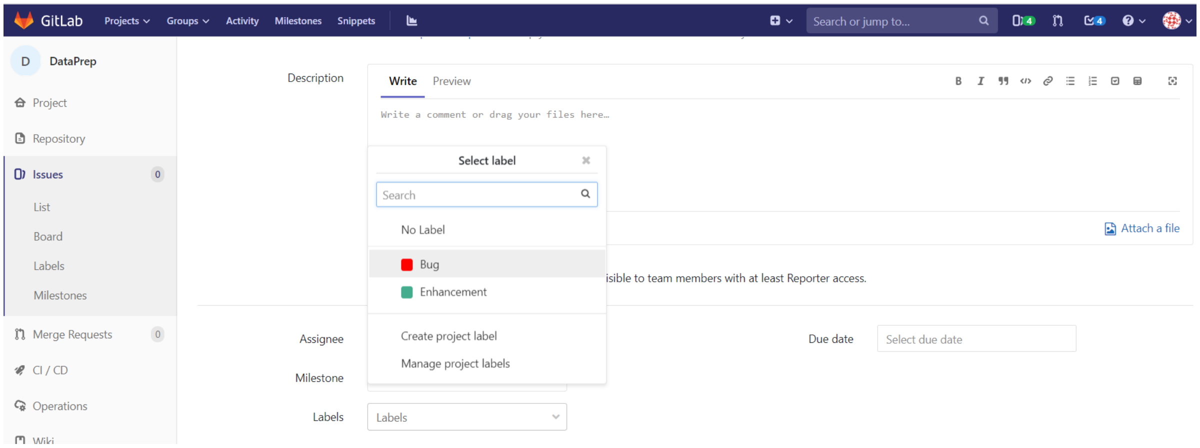


Figure 19: The UI when creating an issue using GitLab.

If you require any clarification or have technical difficulties with the app, please contact Jedid Ahn at jahn@research.baycrest.org and cc both Derek at dbeaton@research.baycrest.org and Kelly at ksunderland@research.baycrest.org.

7 Authorship, contributions, and notes

To come.